Ingénierie des Systèmes d'Information
Vol. 30, No. 3, March, 2025, pp. 647-654
Journal homepage: http://iieta.org/journals/isi

Face Recognition System for Criminal Identification in CCTV Footage Using Keras and OpenCV

Ruchi Rani[1*], Kiran Napte[2], Sumit Kumar[3], Sanjeev Kumar Pippal[4], Megha Dalsaniya[1]

[1] School of Computer Engineering and Technology, Dr.Vishwanath Karad MIT World Peace University, Pune 411038, India
[2] Department of Electronics and Telecommunication, Pimpri Chinchwad College of Engineering and Research, Pune 412101, India
[3] Symbiosis Institute of Technology, Pune Campus, Symbiosis International (Deemed University), Pune 412115, India
[4] GL Bajaj Institute of Technology and Management, Greater Noida 201306, India

Corresponding Author Email: ruchiasija20@gmail.com

https://doi.org/10.18280/isi.300309

Received: 28 July 2024
Revised: 11 November 2024
Accepted: 24 February 2025
Available online: 31 March 2025

Keywords:
face recognition, OpenCV, Keras, deep learning

ABSTRACT

This paper presents an advanced face recognition system for identifying potential criminals using Closed Circuit Television (CCTV) footage. The model is trained using a diverse dataset comprising facial images with variations in age, ethnicity, gender, lighting conditions, and facial expressions (such as smiling, frowning, and wearing glasses). The system leverages deep learning techniques, specifically a Convolutional Neural Network (CNN) with a pre-trained VGG16 architecture integrated with the Keras library for extracting complex facial features. OpenCV is utilized for video preprocessing, frame extraction, and real-time deployment. The model utilizes transfer learning, optimized with the Adam algorithm and a cross-entropy loss function, to enhance its generalization across diverse facial features. The VGG16 model demonstrates strong performance, achieving an accuracy of 97.6%, recall of 96.9%, precision of 97.5%, and an F1-score of 96.9%. This system is designed for real-time surveillance applications, ensuring fast and accurate criminal identification with minimal human intervention.

1. INTRODUCTION

As a result of technology development, deep learning methodologies and computer vision technologies have transformed the range of surveillance and security systems. In this regard, a range of innovations generated from such a combination presents a powerful tool, facial recognition, which can identify individuals across various contexts-from device authentication to law-and-order enforcement. This paper will discuss developing and deploying a state-of-the-art facial recognition system, carefully designed to simplify sorting through vast swaths of CCTV footage to find potential suspects and perpetrators. Facial recognition will harness the capabilities of the VGG16 architecture and the versatile functionality of OpenCV, enhancing traditional surveillance techniques with highly accurate and efficient real-time face detection across the shifting complexities of a world environment.

With such incorporations of deep learning techniques (Keras) and computer vision technologies, the developmental process of surveillance and security systems has accelerated. Among these innovations, facial recognition has been the most outstanding tool in person identification in all contexts, from smartphone authentication to law enforcement. In this study, facial detection is developed and implemented to track possible criminals using CCTV camera footage. Adopting the VGG16 architecture, a pre-trained large-scale image dataset forms the backbone, even under dynamic environmental settings. The integration of OpenCV further enhances its capabilities, allowing for practical deployment in complex surveillance environments [1].

Facial recognition technology (FRT) has played a great role in the identification of suspects. This factor has made the work of the police easier in the sense that they can quickly and efficiently wade through huge amounts of videos to isolate a potential culprit. These are some other real advantages that can be linked with using FRT in investigations, such as minimizing the chances of human mistakes, taking the time for investigations, and generally enhancing the functioning of the police. However, some challenges still come into play in the flow of FRT implementation. One of the main challenges is that the facial appearance may vary based on the camera position, light, and barriers, such as the mask or sunglasses, which creates a challenge for the analyst's work. Concern over demographic biases, which could result in poorer recognition rates for particular groups-particularly those with darker skin tones tend to be lower impacting fairness and raising ethical issues, especially in criminal justice contexts. In addition, some loopholes arise when personifying the FRT models that are more accurate and resistant to real criminal scenarios. Thus, although advances in deep learning have provided more powerful algorithms for facial recognition, further studies of the performance of these systems in the unregulated environment have continued. For these technologies to be

employed widely in crime prevention concerns, FRT has to be enhanced for low resolution and difficult conditions such as lighting or impediments.

Although deep learning has led to more capable facial recognition systems, their effectiveness in uncontrolled environments-like busy public spaces—continues to be a research focus. This work aims to reduce such gaps and enhance the dependability of facial recognition models based on superior accuracy and reduced bias for use in law enforcement. To this end, a new real-time facial recognition system for CCTV videos has been established in this work based on the specially designed dataset. Data acquisition was conducted manually, and each image was labeled by identity, facilitating effective training for the VGG16 model and allowing it to adapt to real-world surveillance scenarios. The dataset was split into a training set (70%) and a test set (30%) for proper model evaluation. For the actual implementation, it was trained with OpenCV for real-time outputs; tests carried out on actual video flow confirmed the efficiency and consistency of the system, suggesting its potential implementation in real security applications. Creating a real-time facial recognition system only for CCTV is a breakthrough in using artificial intelligence in surveillance. The proper selection of datasets, optimization for transfer learning, and incorporation of OpenCV functions prove that the system possesses the stability to treat images and, namely, faces in real-life environments successfully [2, 3]. Testing in actual environments reinforces its potential for security applications, offering law enforcement a powerful tool to improve public safety [4]. This work contributes to developing and refining facial recognition systems and points toward the future of smart surveillance. As AI technology advances, its role in community safety will likely grow, with enhanced facial recognition playing a major role in preventing crime and supporting law enforcement. This investigation assists development and optimization by laying the groundwork for a future where intelligent systems will play a crucial role in safeguarding communities and ensuring law and order [5]. This paper covers the following:

- A Deep dive into the principles and mechanics behind facial recognition algorithms, specifically focusing on the VGG16 architecture implemented using Keras and OpenCV.
- A review of the data preparation techniques and approaches employed in executing transfer learning while training facial models for criminal identification.
- Test the system using actual CCTV recordings to assess its real-world performance in identifying suspects, offering insights into the system’s practical for law enforcement.

2. LITERATURE REVIEW

The study [1] presents a confront acknowledgment calculation utilizing OpenCV and the PCA algorithm. It traces the confront acknowledgment prepared, including confronting location, representation, and acknowledgment. The PCA calculation is utilized for acknowledgment, utilizing Visual Studio to construct a framework that identifies faces, clarifies highlights, and conducts acknowledgment preparation. The pre-processing phase involves estimating mean and difference faces, constructing a covariance matrix, extracting eigenfaces, and projecting faces for recognition. The identification phase transforms the face into a feature space. This necessitates face location, representation, and identification for biometric technology, which has concentrated on face recognition instead of face detection because it is more effective or elaborate. The investigation gives a detailed specialized establishment for understanding PCA, emphasizing cruelty, fluctuation, and standard deviation in factual investigation, and concludes with effective confront acknowledgment re-enactments illustrating the algorithm's viability in character confirmation. A face recognition system developed using OpenCV and Python is introduced to address security access control issues and the attendance monitoring needs of an organization [2]. It utilizes the Haar Cascade classifier with OpenCV to achieve efficient face detection within an image or video feed. Once faces are found, the framework extricates facial highlights and changes them into numerical representations called embeddings. At that point, these embeddings are utilized to prepare a machine learning demonstration, particularly the OpenCV nearby double-designed histograms (LBPH)recognizer. In recognition of facial images, modern face embeddings are extracted and matched against embeddings stored in a trained model. If a close match is found, the person in the picture is identified by this model. This method is convenient, cheap, and requires minimal effort as it is based on OpenCV's open-source features and Python's simplicity. However, the researchers suggest exploring sophisticated algorithms in the future for handling novel situations with multiple faces or faces obscured by accessories, such as spectacles or caps.

Security frameworks utilize OpenCV and Python for facial acknowledgment, particularly about challenges caused by various lighting conditions [3]. The framework leverages OpenCV's LBPH calculation, known for its flexibility in light changes. During the preparation stage, LBPH extricates neighborhood highlights from facial pictures within the dataset and changes them into histograms, which act as compact representations of these highlights. When an unused confrontation is displayed to the framework within the acknowledgment stage, LBPH extricates comparative highlights and produces a histogram. This recently created histogram is at that point compared to the histograms put away within the prepared show. The framework recognizes the individual within the picture if a near match is found based on likeness measurements. The system has more than 80% precision in recognizing permitted faces, illustrating its adequacy in recognizing people gathered to be there and potential trespassers. Within the current usage of the framework, identifying protest area highlights or administrations are basic for upgraded security measures. When the framework joins address disclosure, it does not, as it were, distinguish true blue faces but to other objects inside the border, possibly moving forward security through cautioning on unauthorized objects or actions. The study [4] examines how OpenCV and Python could be used as tools in departmental confrontation. It discusses both the fundamentals and its application modes. The study focused on OpenCV’s capability to recognize faces and provided a guide for building up applications systematically. Although the mentioned advantages may include practical use and high-level security, there is a need for further improvement. Although this system may have certain advantages in terms of counting possibility and advancing protection, observance maintained by the paper requires a detailed examination of what strategies are currently being applied to recognize any challenges and the specific

techniques used in building it. The study [5] investigates how OpenCV works as a computer vision stage that upgrades the facial acknowledgment framework. It digs into the fundamental standards of facial acknowledgment innovation, sketching out its employments and instances where OpenCV was utilized to realize these targets. The paper plunges into the points of interest of the Haar cascade calculation, a broadly utilized strategy for confronting location inside pictures. Whereas the clarification covers the framework setup, libraries required, and the Haar cascade calculation, a more comprehensive investigation of highlight extraction procedures, which play a pivotal part in recognizing faces, would upgrade the paper's commitment. The locator, information set maker and coach modules' execution points of interest are not clearly explained; this makes it hard to understand the entire system's design. The evaluation approach is limited and only reflects effective face detection from a distance of 200cm. This article investigates how OpenCV is a computer vision stage that upgrades the facial acknowledgment framework. It digs into the fundamental standards of facial acknowledgment innovation, sketching out its employments and instances where OpenCV was utilized to realize these targets.

3. METHODOLOGY

This study focuses on developing an innovative facial recognition method that utilizes the strengths of both OpenCV and Keras [5-10]. OpenCV has high-quality features and is fine-tuned to perform fast and efficient facial detection tasks. The method implements robust feature extraction and recognition through the VGG16 model by incorporating deep learning techniques in Keras. This combined approach offers greater precision and effectiveness than conventional strategies. Figure 1 describes a facial recognition system that uses video input for real-time face detection and criminal recognition, utilizing a pre-trained VGG16 model.

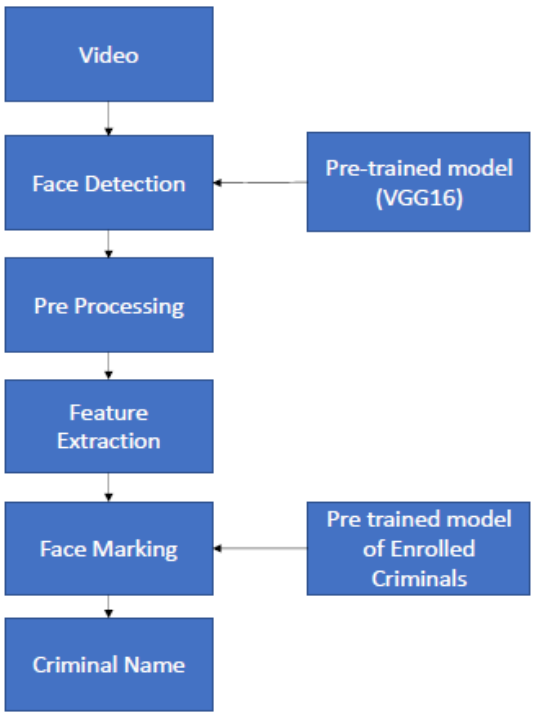

Figure 1. Block diagram of face recognition

3.1 Collection of dataset and data preprocessing

The framework persistently captures video information, usually fetched from a camera source. This stream of frames is a sequence containing a momentary description of the scene under watch. The number of frames captured per second depends on the frame rate, and a higher rate leads to smoother videos. This show has been particularly planned for the assignment of confronting discovery. Amid its preparation, the show has uncovered a gigantic dataset of assorted facial pictures, including various varieties. These varieties incorporate facial highlights from individuals of distinctive ages, ethnicities, sexual orientations, and indeed facial expressions (grinning, scowling, with glasses, etc.). The show has learned to recognize the designs and characteristics that recognize faces from other objects inside a picture. When handling an outline, the show filters each location of the picture, analyzing the pixel designs and applying information about facial highlights. If the demonstrate recognizes a locate with a tall likelihood of containing a confront, it yields a bounding box around the recognized facial locale. This bounding box indicates the area and degree of the confront inside the outline.

3.2 Video input and face detection

The framework persistently captures video information, usually fetched from a camera source. This stream of frames represents a real-time description of the scene being unmonitored. The number of frames captured per second depends on the frame rate, and a higher rate leads to smoother videos. The VGG16 model, pre-trained on a diverse dataset of facial images, was designed for face detection tasks. The variation incorporated in the dataset for this study is based on age, ethnicity, gender, and mood, in which some depict smiling while others frown or wear pressed glasses. The model has been taught to differentiate between faces from other forms since their nature differs. When handling a frame, VGG16 examines the areas, learning from the previously seen pixel patterns and understanding faces. If the model finds a portion of the image that probably contains a face, it outlines this portion with a rectangular box to indicate the location and extent of the face in a particular frame.

3.3 Pre-processing and feature extraction

When working with such images, one has to normalize pixels to achieve better results when processing images. Therefore, A pixel value can fluctuate from one scene or shot to another depending on factors such as light quality, settings, etc. Normalization scales such values between 0 and 1 or -1 and 1 depending on data distribution to make the model more generalized across frames. One may include the following strategy: converting images from RGB format to grayscale [11-15], which makes them structurally similar for defining features. Once the images are prepared, the VGG16 model comes into play and acts as a feature extractor. That is, it works only with the regions of the image that were determined to contain faces. As it will be seen through these layers, the model continues to enhance its analysis of the image during its processing. The first layers integrate simple details like the edge or line, but the next layers observe even more detail about the face and its important facets. This allows the model to focus on features such as the relative position of the eyes, the

nose, and the mouth or even marks and wrinkles, which are vital for identification purposes.

3.3 Face marking and criminal name

The system successfully identifies the individual once the extracted features are matched with a known criminal in the database (with confidence exceeding a pre-defined threshold). The output consists of two components: a visual cue (e.g., bounding box or highlight) around the detected face and an overlay displaying the criminal's name. This enhances the identification process by providing both visual confirmation and contextual information. This facial recognition framework uses the pre-trained VGG16 model for real-time face detection, feature extraction, and criminal identification within a continuous video stream. The system can then match the extracted features to that of a database of criminals, and if the match scores are high enough, the system can positively identify an individual.

4. SYSTEM ARCHITECTURE

Figure 2 shows the system architecture diagram explained in the subsequent sections.

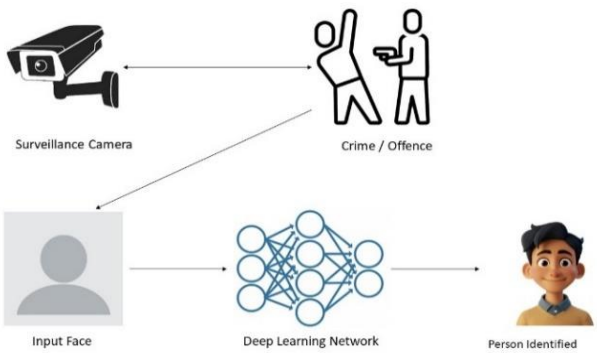

Figure 2. System architecture diagram

- **Surveillance Camera and Crime Detection**

It starts with identifying a real-time surveillance camera that records live events of either a public or private venue. The camera is on, and the video recording is constantly running and may display images of a crime or an offense being committed. The tapes are the source input as the facial recognition system will process the recorded videos.

- **Crime/Offense Event**

As seen in the surveillance video, when a criminal incident occurs, such as theft, assault, or any form of violation, the system can detect people in the scene. The system concentrates on obtaining the facial characteristics of those present during or after the crime. Input Face: The next operation separates the face of the suspect or the persons of interest from the CCTV [16-20] feed. It can also be operated manually, where an operator draws a face or operated automatically using face detection software. After identifying the face from the frame, the frame is provided as an input image to the deep learning network.

- **Deep Learning Network**

The detected face goes through a classification process fed to the VGG16 [21-24] network, a sort of deep learning. which is intended for face recognition. The network selects particular features of the input face containing such peculiarities as different facial contours found in structural plans, block and open-grounded profiles, and textural profiles. They are then compared with a mapped database of individuals that can be checked, such as criminal records or certain suspects. To minimize these problems, using a multi-layered VGG16 model contributes greatly to the real-time recognition of faces and accurate matching of features.

Input Layer Dimensions: The input images should be 224×224 pixels, and images should be expected to belong to three color channels. which are Red, Green, and Blue. This standard size allows the model to process images quickly without compromising a lot of memory.

Convolutional Blocks: Six layers of VGG16 have a basic configuration of convolution and max pooling with one or two layers in each of the five convolutional blocks. Essentially, each convolutional layer shown in Figure 3 performs a certain set of filters to the input data and outputs several features from the images.

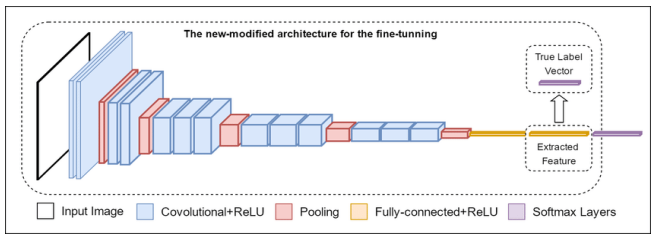

Figure 3. Architecture of VGG16

- **Block 1:** Layers: 2 Convolutional layers. Filters: Each layer has 64 filters. Activation Function: ReLU (Rectified Linear Unit) sometimes includes non-linearity. Pooling: A MaxPooling layer comes next, which cuts the dimensions of the feature maps in two but retains only the values most relevant to the output.
- **Block 2:** Layers: 2 Convolutional layers. Filters: Each layer has 128 filters. Activation Function: ReLU. Pooling: This is then topped up with a MaxPooling layer which again brings down the dimension.
- **Block 3:** Layers: 3 Convolutional layers. Filters: Each layer has 256 filters. Activation Function: ReLU. Pooling: Then, a MaxPooling layer is used to reduce the size of the matrix.
- **Block 4:** Layers: 3 Convolutional layers. Filters: Each layer has 512 filters. Activation Function: ReLU. Pooling: A MaxPooling layer is used to down-sample and reduce the dimensions.
- **Block 5:** Layers: 3 Convolutional layers. Filters: Each layer has 512 filters. Activation Function: ReLU. Pooling: A MaxPooling layer is followed to reduce the dimensions further.

Flattening Layer Before the nearest neighbor sampling layer, a Flattening layer changes the 3d feature maps to a 1D vector. The amount of supernode features in the output size of this vector is 25088 (from a 7×7×512 feature map). Fully Connected Layer The last fully connected layer is a Dense layer with four neurons corresponding to the count of criminal categories to be predicted. The Softmax activation function is applied here to transform the output logits to the probabilities of each class.

- **Person Identified**

Upon decision-making from the deep learning network, the

system records a match within the database and certifies the person's identity. The output is the identity of the person, which can be an associated name, record number, or any other identification number. This identification will, therefore, assist every security or police force in easily identifying and apprehending the perpetrator of the crime.

5. ALGORITHM OF THE PROPOSED MODEL

The following pseudocode outlines the key steps involved in training the VGG16 model for facial recognition:

START
// Step 1: Set Up the Project
createProject()
- Initialize a new project for face recognition.
- Set up directories for data (train and test datasets) and code files.
- Install required libraries like TensorFlow and Keras.
Command:
pip install tensorflow keras
OUTPUT: Local project directory with required dependencies installed.

// Step 2: Define Image Dimensions and Paths
defineImageParams()
- Set the image dimensions to (224, 224) to match VGG16 input.
- Define paths for training and validation datasets.
OUTPUT: Image size and dataset paths defined.

// Step 3: Load Pre-trained VGG16 Model
loadPretrainedModel()
- Load VGG16 model with input shape of (224, 224, 3).
- Exclude the top layer (pre-trained classifier) to add custom layers.
OUTPUT: Pre-trained VGG16 model loaded without the top classifier.

// Step 4: Freeze Pre-trained Layers
freezeModelLayers()
- Freeze all layers of the VGG16 model to retain their pre-trained weights.
OUTPUT: VGG16 layers frozen for feature extraction.

// Step 5: Add Custom Layers
addCustomLayers()
- Add a Flatten layer to flatten the VGG16 output.
- Add a Dense layer with softmax activation for classification.
OUTPUT: Custom layers added to the model.

// Step 6: Compile the Model
compileModel()
- Compile the model using 'categorical_crossentropy' as the loss function.
- Use 'adam' optimizer for efficient training.
OUTPUT: Model compiled and ready for training.

// Step 7: Data Augmentation
augmentData()
- Apply data augmentation to the training set using scaling, shearing, zooming, and flipping.
- Rescale validation data without augmentations.
OUTPUT: Training and validation data generators created with augmentation for training data.

// Step 8: Train the Model
trainModel()
- Fit the model on the training dataset.
- Validate the model using the validation dataset.
- Set epochs to 10 for quick training.
OUTPUT: Model trained on augmented data and validated.
evaluateModel()
- Evaluate the model on the validation/test dataset.
- Calculate accuracy, precision, recall, and F1-score.

// Step 9: Plot Loss and Accuracy
plotMetrics()
- Plot training and validation loss over epochs.
- Plot training and validation accuracy over epochs.
OUTPUT: Training and validation metrics plotted for analysis.

// Step 10: Save the Model
saveModel()
- Save the trained model to a file.
OUTPUT: Model saved as 'facefeatures_new_model.keras'.

// Step 11: Perform Real-time Face Recognition
realTimeRecognition()
- Load the saved model.
- Capture frames from webcam.
- Preprocess frames (resize to 224x224, normalize).
- Run face recognition on each frame.
IF face is recognized:
- Display label with the person's name.
ELSE:
- Display "Unknown".
OUTPUT: Real-time face recognition displayed on webcam feed.

// Step 12: Optimize for Performance
optimizePerformance()
- Minimize latency in real-time recognition.
- Reduce input frame size or modify model to speed up processing.
OUTPUT: Optimized real-time face recognition system.

6. RESULTS AND DISCUSSION

The model was trained using a learning rate of 0.001, which ensures smooth convergence with the Adam optimizer. A batch size of 32 was selected to balance memory efficiency and training speed, while the number of epochs was set to 10 to allow effective learning without overfitting. The model was evaluated on a test set. The performance was measured using accuracy, precision, recall, and F1-score, with the results summarized below in Table 1.

The following results were obtained after testing our model on the validation set. Figure 4 shows the training accuracy and validation accuracy of the VGG16 model, and Figure 5 shows the training and validation loss of the VGG16 model. These graphs demonstrate the model's performance and ability to

generalize effectively during the training and validation phases. As shown in Table 2, Our model outperforms ResNet50, InceptionV3, and MobileNetV2, achieving higher accuracy, especially in complex scenarios. With a remarkable accuracy of 97.6%, VGG16 demonstrates superior performance in handling variations such as lighting, angles, and object orientations. Its high precision and recall make it a more reliable choice for tasks requiring detailed image recognition, ensuring consistent performance in real-world applications like autonomous vehicles and surveillance systems.

Table 1. Performance parameters

Metric	Value
Accuracy	97.2
Precision	80
Recall	80
F1-Score	79

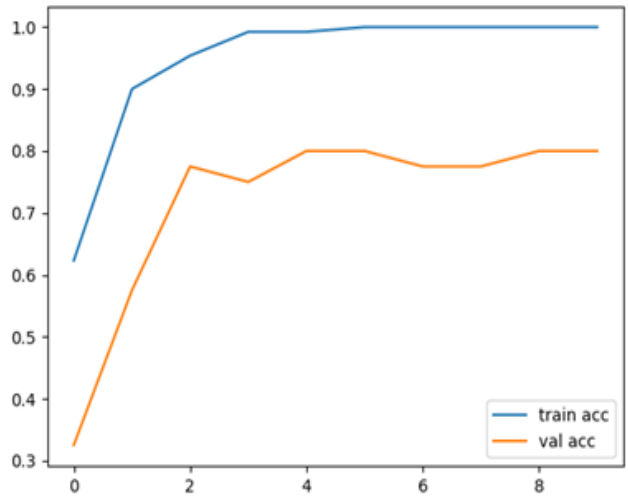

Figure 4. Training accuracy

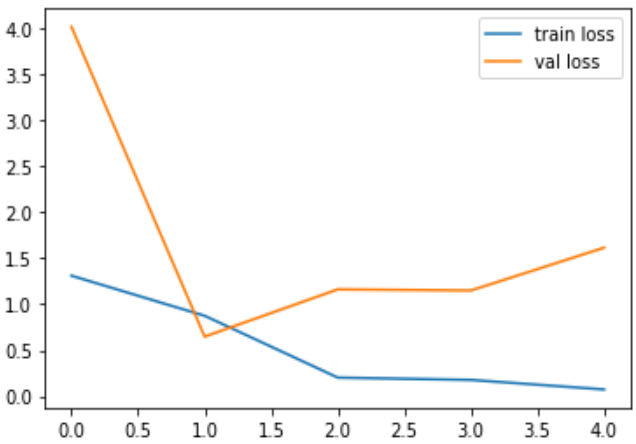

Figure 5. Training loss

Table 2. Comparison of different models

Model	Accuracy	Precision	Recall	F1-Score
ResNet50	91.4%	90.4%	92.8%	91.6%
InceptionV3	91.5%	91.4%	91.5%	91.8%
MobileNetV2	90.0%	91.5%	92.3%	92.5%
VGG16	97.6%	97.5%	96.9%	96.9%

7. POTENTIAL SOCIAL IMPACTS OF FACIAL RECOGNITION TECHNOLOGY

Facial recognition technology, if utilized in tracking criminals, would greatly boost public safety and the effectiveness of law enforcement. It would go a long way in preventing crimes and speeding up investigation in cases by automating the identification processes of suspects in big video datasets like CCTV footage. It could add valuable evidence, thus aiding law enforcers in solving cases. It's application in risky public places can point out people under surveillance, thus offering a way of reducing security risks. However, these benefits are to be weighed against privacy issues, especially when citizens may feel assaulted by constant surveillance in public areas or lose their right to privacy through tracking without consent. Therefore, the technology should be used appropriately and openly so as not to betray public trust. Even though it has some merits, facial recognition does have its issues and controversies surrounding possible bias and discrimination. In many cases, the system readily gives high error identification rates for certain subjects from minority groups that may be used against them or even to discriminate against them. False positives or misidentifications may cause wrongful arrests, damaged reputations, and loss of credibility of the system. However, the surveillance will also intrude on civil liberties and undermine some of the fundamental rights of citizens by suppressing free expression and sowing a climate of fear. High levels of privacy regulation, reducing bias, and responsible use of face recognition technology will mitigate adverse consequences.

8. ETHICAL ISSUES AND PRIVACY PROTECTION MEASURES IN FACIAL RECOGNITION TECHNOLOGY

This technology holds incredibly strong powers in preventing crimes and identification but poses very serious ethical issues; they range from privacy to consent and the security of data gathered. The biggest challenge lies in its intent to misuse it, such as mass surveillance without the public's knowledge or consent. In this case, the greatest challenge of surveillance is that most people may not even know they are under surveillance, thus indirectly infringing on their right to privacy. In some instances, they do not know they are being followed or the use of their data. Furthermore, this technology may allow followers to go from public spaces to private ones, thus creating enormous private databases containing personal information that can easily be misused if there is no proper protection strategy. Such issues should be answered with measures for sheltering privacy and how the technology will be responsibly applied. For example, facial data collection and usage must strictly follow laws regarding data protection, such as the General Data Protection Regulation, taking transparency and accountability as key and getting clear, informed consent from the person whose data is being culled. However, facial recognition systems should also encrypt their data using robust encryptions so that it is not accessed without authorization and interfered with by cyber threats. These systems must be audited as often as required to ensure that ethics are present, and their services are used reasonably without discrimination and in line with all privacy laws.

9. CONCLUSION

Our developed face recognition system employs Keras and OpenCV as a powerhouse tool in criminal identification. This is done by applying the VGG16 architecture type CNN, thereby enabling successful extraction and processing of features that appear on the faces of criminals and comparing known faces with new ones to make tentative identifications of suspects. While very good at performance, with VGG16 featuring an astonishingly high accuracy rate of 97.6%, this system's limitations and ethical considerations are something to pay heed. As much of the accuracy of the system hinges on the quality and diversity of training data, changes or otherwise posed factors like lighting can impair its operations in detecting faces. Hence, the work already done on data augmentation and pose-invariant methods can be further considered to increase the system's consistency in activity recognition. Along with this, proper development is needed to understand privacy issues and, probably, the issues of bias. It should ensure legal and ethical compliance standards. Once these issues are met, face recognition systems like this will be valuable assets in criminal investigations.

REFERENCES

[1] Tan, L., Wu, F., Yin, X., Liu, W. (2021). Face recognition algorithm based on open CV. In 2021 6th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, pp. 96-100. https://doi.org/10.1109/CCISP52774.2021.9639288

[2] Singh, G., Gupta, I., Singh, J., Kaur, N. (2022). Face recognition using open source computer vision library (OpenCV) with Python. In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), Noida, India, pp. 1-6. https://doi.org/10.1109/ICRITO56286.2022.9964836

[3] Mitra, D., Gupta, S., Goyal, A. (2022). Security system using open cv based facial recognition. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 371-374. https://doi.org/10.1109/ICACITE53722.2022.9823881

[4] Dhawle, T., Ukey, U., Choudante, R. (2020). Face detection and recognition using OpenCV and Python. International Research Journal of Engineering and Technology (IRJET), 7(10): 1269-1271.

[5] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-TR06', Technical Report, Erciyes University.

[6] Li, L., Mu, X., Li, S., Peng, H. (2020). A review of face recognition technology. IEEE Access, 8: 139110-139120. https://doi.org/10.1109/ACCESS.2020.3011028

[7] Sirivarshitha, A.K., Sravani, K., Priya, K.S., Bhavani, V. (2023). An approach for face detection and face recognition using OpenCV and face recognition libraries in Python. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1274-1278. https://doi.org/10.1109/ICACCS57279.2023.10113066

[8] Barnouti, N.H., Al-Mayyahi, M.H.N., Al-Dabbagh, S.S.M. (2018). Real-time face tracking and recognition system using Kanade-Lucas-Tomasi and two-dimensional principal component analysis. In 2018 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, pp. 24-29. https://doi.org/10.1109/ICOASE.2018.8548818

[9] Ramadhani, A.L., Musa, P., Wibowo, E.P. (2017). Human face recognition application using pca and eigenface approach. In 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, Indonesia, pp. 1-5. https://doi.org/10.1109/IAC.2017.8280652

[10] Zhang, M., Liao, W., Zhang, J., Gao, H., Wang, F., Lin, B. (2019). Embedded face recognition system based on multi-task convolutional neural network and LBP features. In 2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE), Fuzhou, China, pp. 132-135. https://doi.org/10.1109/ICIASE45644.2019.9074104

[11] BenSaid, E., Neji, M., Jabberi, M., Alimi, A.M. (2025). Deep keypoints adversarial attack on face recognition systems. Neurocomputing, 621: 129295. https://doi.org/10.1016/j.neucom.2024.129295

[12] Mamun, K.A., Nabid, R.A., Pranto, S.I., Lamim, S.M., Rahman, M.M., Mahammed, N., Huda, M.N., Sarker, F., Khan, R.R. (2024). Smart reception: An artificial intelligence driven Bangla language based receptionist system employing speech, speaker, and face recognition for automating reception services. Engineering Applications of Artificial Intelligence, 136: 108923. https://doi.org/10.1016/j.engappai.2024.108923

[13] Ghani, M.A.N.U., She, K., Rauf, M.A., Khan, S., Khan, J.A., Aldakheel, E.A., Khafaga, D.S. (2024). Enhancing security and privacy in distributed face recognition systems through blockchain and GAN technologies. Computers, Materials and Continua, 79(2): 2609-2623. https://doi.org/10.32604/cmc.2024.049611

[14] Jegadeesan, S., Monika, M., Oviya, P., Supriya, M. (2022). Artificial intelligence based face recognition using deep learning. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1017-1024. https://doi.org/10.1109/ICOEI53556.2022.9776744

[15] Sardar, A., Umer, S., Rout, R.K., Pero, C. (2023). Face recognition system with hybrid template protection scheme for Cyber-Physical-Social Services. Pattern Recognition Letters, 174: 17-24. https://doi.org/10.1016/j.patrec.2023.08.011

[16] Zhang, J., Yi, Q., Lu, D., Sang, J. (2023). Low-mid adversarial perturbation against unauthorized face recognition system. Information Sciences, 648: 119566. https://doi.org/10.1016/j.ins.2023.119566

[17] Jagtap, S., Chopade, N.B., Tungar, S. (2022). An investigation of face recognition system for criminal identification in surveillance video. In 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, Pune, India, pp. 1-5. https://doi.org/10.1109/ICCUBEA54992.2022.10010987

[18] Ratnaparkhi, S.T., Tandasi, A., Saraswat, S. (2021). Face detection and recognition for criminal identification system. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, pp. 773-777. https://doi.org/10.1109/Confluence51648.2021.9377205

[19] Wang, M.J. (2018). Face feature dynamic recognition method based on intelligent image. In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Hunan, China, pp. 57-60. https://doi.org/10.1109/ICVRIS.2018.00022

[20] Railkar, Y., Pawar, S., Pise, R., Nasikkar, A., Patil, P. (2024). Criminal recognition system. In International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, pp. 1-6, https://doi.org/10.1109/ESCI59607.2024.10497321

[21] Kumar, S., Rani, R., Chaudhari, U. (2024). Real-Time sign language detection: Empowering the disabled community. MethodsX, 13: 102901. https://doi.org/10.1016/j.mex.2024.102901

[22] Dodia, A., Kumar, S. (2023). A comparison of yolo based vehicle detection algorithms. In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, pp. 1-6. https://doi.org/10.1109/ICAIA57370.2023.10169773

[23] Rani, R., Kumar, S., Pippal, S.K., Gund, M., Chaudhari, U., Agrawal, R., Dalsaniya, M., Verma, L. (2024). Ips: Intelligent parking system using YOLO and image processing. International Journal of Transport Development and Integration, 8(3): 447-453. https://doi.org/10.18280/ijtdi.080308

[24] Kumar, S., Rani, R., Pippal, S.K., Chaudhari, U. (2024). Real time Indian sign language recognition using transfer learning with VGG16. TELKOMNIKA (Telecommunication Computing Electronics and Control), 22(6): 1459-1468. http://doi.org/10.12928/telkomnika.v22i6.26498